

Remarks

Claims 1-26 are pending in this application. Claims 1-26 have been rejected. No claims have been amended by this Response.

Claim Rejections: 35 USC § 102

In the above-referenced Office Action, claims 1, 9, 17, and 25-26 were rejected under 35 U.S.C. § 102(e) as being anticipated by Alpern et al., U.S. Patent No. 6,470,361 B1. Alpern, however, discloses a generational garbage collection scheme in which objects are collected based on their age (Alpern, col. 2, line 56 - col. 3, line 6; col. 5, line 34-43). Claims 1, 9, 17 and 25-26 of the present application, in contrast, are directed to garbage collection based on object type. Claims 1, 9, 17 and 25-26 of the present application, therefore, patentably distinguish over Alpern.

More specifically, Alpern is directed to techniques for use in a generational garbage collection scheme (col. 5, lines 21-30). As Alpern describes, "[g]enerational schemes partition the objects in the heap into groups called 'generations,' based upon the ages of the objects, where an object's age is typically measured in terms of the number of garbage collections that the object has survived" (emphasis added) (col. 2, line 65 - col. 3, line 2). The Summary section of Alpern, for example, makes clear that the disclosure in Alpern is directed to "the efficient management of remembered sets

*in a generational garbage collection scheme" (emphasis added) (col. 5, lines 33-35).* The passages cited in the above-referenced Office Action describe partitioning of objects based on the age of the objects (col. 3, lines 23-65; col. 5, lines 33-67; col. 6, lines 1-26).

In contrast, claim 1 of the present application recites "an object allocation routine which stores an object of a particular type in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object type" (emphasis added). The specification of the present application discloses the "char[]" object type in the Java™ programming language as one example of an object type (page 8, line 23 through page 9, line 5).

The object allocation routine of claim 1 stores an object of a particular type in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object type. In other words, the object allocation routine of claim 1 performs partitioning based on object type, not based on object age. Alpern, in contrast, discloses techniques for partitioning objects based on object age, not based on object type. Alpern therefore does not disclose the object allocation routine recited in claim 1 of the present application. For at least this reason, claim 1 of the present application patentably distinguishes over Alpern.

The same reasoning applies to claims 9, 17, and 25-26 of the present application. Claim 9, for example, recites "means for storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object *type*" (emphasis added). Claim 17 recites a step of "storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object *type*" (emphasis added). Claim 25 recites a collector "storing an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the *type*" (emphasis added). Finally, claim 26 recites program code which "stores an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the *type*" (emphasis added). Claims 9, 17, and 25-26 therefore patentably distinguish over Alpern for at least the same reasons as claim 1.

The above-referenced Office Action states that the Applicant's arguments provided in the Response dated May 21, 2004 have been fully considered but that they are not persuasive. Applicant previously argued, and continues to assert, that Alpern does not disclose "an object allocation routine which stores an object of a particular *type* in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object *type*" (emphasis added), as required by claim 1 of the present application.

The above-referenced Office Action fails to point to any disclosure in Alpern which teaches or suggests this limitation of claim 1.

For example, the above-referenced Office Action at page 8 states that:

it is well-known that data structure in object oriented database programming assign object type that can be user defined in reference to the objects in an object class. Alpern teaches object classes which are categories of objects that can be grouped together by object type.

These statements appear to assert that an object class defines a category of objects which share the same object type. Even assuming for purposes of argument that this assertion is both true and disclosed by Alpern, such a disclosure does not teach or suggest the above-referenced limitation of claim 1 of the present application, namely "an object allocation routine which stores an object of a particular type in one of a plurality of logical partitions in the heap *dependent on a predefined category assigned to the object type*" (emphasis added). The mere existence of objects sharing a common object type does not teach or suggest techniques for performing garbage collection based on object type in the manner recited by claim 1. In fact, Alpern does not disclose techniques

for performing garbage collection based on object type. Rather, Alpern discloses garbage collection based on age.

In summary, Alpern fails to disclose an express limitation of claim 1 of the present application, namely "an object allocation routine which stores an object of a particular type in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object type". Claim 1 of the present application therefore patentably distinguishes over Alpern. Claims 9, 17, and 25-26 of the present application patentably distinguish over Alpern for at least the same reasons. Applicant therefore respectfully requests that the rejection of claims 1, 9, 17, and 25-26 of the present application be withdrawn.

Claim Rejections: 35 USC § 103

Claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Alpern et al., U.S. Patent No. 6,470,361 B1, in view of Engelstad et al., U.S. Patent No. 5,485, 613. As described above, Alpern does not disclose techniques for performing garbage collection based on object type, as required by independent claims 1, 9, 17, and 25-26 of the present application. Furthermore, the above-referenced Office Action points to no teaching or suggestion in Engelstad for performing garbage collection based on object type, and Engelstad in fact provides no such teaching or suggestion. Because neither Alpern nor Engelstad

teach or suggest performing garbage collection based on object type, the combination of Alpern and Engelstad provides no such teaching or suggestion. Dependent claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24, therefore, patentably distinguish over the combination of Alpern and Engelstad for at least the reasons provided above with respect to independent claims 1, 9, 17 and 25-26. Applicant therefore respectfully traverses the rejection of claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24, and requests that the rejection thereof be withdrawn.

The above-referenced Office Action states at page 8 that:

Engelstad in column 16, lines 18-55 goes further and specifically teach each generation 630 is defined by a data structure of the type represented by Table 2. . . .

The Examiner respectfully points to Engelstad column 16, lines 18-55, wherein each generation 630 is defined by a data structure of the type represented by Table 2, is disclosed.

Engelstad also teaches a data structure is used to identify the objects in the generation, and the generation itself.

The Office Action, however, fails to indicate how Table 2 of Englestad or any of the surrounding text teaches or suggests the

limitations of claim 1 of the present application. In fact, Table 2 of Engelstad, and the surrounding text, provides no such teaching or suggestion. Rather, col. 16, lines 18-54, which includes Table 2, specifically describes the performance of *generational* garbage collection - garbage collection based on age - not garbage collection based on type. The data structure represented by Table 2 "is used to identify the objects *in the generation, and the generation itself*" (emphasis added) (col. 16, lines 22-23). In the generation-based garbage collection scheme described in this portion of Engelstad, "objects are grouped into roughly contemporaneous generations as they are created" (col. 16, lines 20-22). The generational information contained in the data structure represented by Table 2, therefore, is used to perform garbage collection based on the *generation* (age), not the type, of objects.

Although the word "type" is used on line 30 of col. 16 of Engelstad, this refers to the type of the data structure represented by Table 2 (col. 16, lines 29-31), not to the type of the objects on which garbage collection is performed. Engelstad does not disclose performing garbage collection on the data structure represented by Table 2 based on its type. Rather, Engelstad describes using the data structure represented by Table 2 to perform garbage collection on other objects based on their age. The mere use of the word "type" in the passage quoted by the Office Action does not teach or suggest the use of object type as a basis for garbage collection.

The above-referenced Office Action further states at pages 8-9 that:

Engelstad in column 15, lines 60-67, and column 16, lines 1-17 goes further to disclose references associated with class objects and object descriptors, which is interpreted by the examiner to read on object types. According to www.dictionary.com, object type is defined as a type is a classification of data that tells the compiler or interpreter how the programmer intends to use it. Alpern in column 2, lines 1-10, prior art, teaches object class and data definitions. Therefore, the combination of Alpern and Engelstad does teach performing garbage collection based on object type.

The above-cited passages in Engelstad do not teach or suggest "an object allocation routine which stores an object of a particular type in one of a plurality of logical partitions in the heap dependent on a predefined category assigned to the object type" (emphasis added), as recited in claim 1 of the present application. Rather, col. 15, lines 40-58 of Engelstad merely describe objects having object headers including the data in Table 1. This passage does not teach or suggest performing garbage collection based on the

type of such objects. Col. 15, line 59 through col. 16, line 17 merely describes the contents of the object header of Table 1 in more detail. This passage does not, however, teach or suggest performing garbage collection based on the object type of objects having the header of Table 1. Indeed, the text that immediately follows (col. 16, lines 18-54) describes generational garbage collection. The above-referenced portion of Engelstad, therefore, provides no teaching or suggestion of the relevant limitations of claim 1 of the present application. Therefore, the combination of Alpern and Engelstad fails to provide a basis for rejecting claim 1. of the present application.

Rejected dependent claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24 contain or incorporate the above-referenced limitation of claim 1 or substantially similar limitations. Because the combination of Alpern and Engelstad fails to teach or suggest this limitation for the reasons stated above, claims 2-3, 6-8, 10-11, 14-16, 18-19, and 22-24 patentably distinguish over the combination of Alpern and Engelstad for at least the same reasons. Applicant therefore respectfully traverses the rejection of these claims and requests that the rejection thereof be withdrawn.

Concluding Remarks

Any dependent claims not specifically referenced above incorporate the limitations of the independent claims from which they depend, and therefore are patentable for at least the same reasons.

If the Examiner considers the arguments presented herein not to be persuasive, the Applicant respectfully requests that the Examiner contact the Applicant to schedule an interview at a mutually convenient time.

If this response is not considered timely filed and if a request for extension of time is otherwise absent, applicant hereby requests any extension of time. Please charge any fees or make any credits, to Deposit Account No. 08-2025.

Respectfully submitted,



Robert Plotkin, Esq.

Reg. No. 43,861



Date

Robert Plotkin, P.C.  
45 Butternut Circle  
Concord, MA 01742-1937  
Tel: (978) 318-9914  
Fax: (978) 318-9060